(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number : **0 469 850 A2**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number : **91306963.9**

(51) Int. Cl.^{5} : **G06F 9/44**

(22) Date of filing : **30.07.91**

(30) Priority : **30.07.90 US 560645**

(43) Date of publication of application :
05.02.92 Bulletin 92/06

(84) Designated Contracting States :
DE GB

(71) Applicant : **Epard, Marc A.**
3104 W. 38th
Lawrence, Kansas 660247 (US)
Applicant : **Denneller, Charles Ross**
919 Alma Drive
Lawrence, Kansas 66047 (US)

(72) Inventor : **Epard, Marc A.**
3104 W. 38th
Lawrence, Kansas 660247 (US)
Inventor : **Denneller, Charles Ross**
919 Alma Drive
Lawrence, Kansas 66047 (US)

(74) Representative : **Colgan, Stephen James et al**
CARPMAELS & RANSFORD 43 Bloomsbury Square
London WC1A 2RA (GB)

(54) **Method and apparatus for pre-recording, editing and playing back presentations on a computer system.**

(57) A system for pre-recording, editing, and playing back a presentation on computer systems creates "tapes" of screen session files, marks and edits them, records and synchronizes sound on them, annotates them with buttons or other overlay graphics, presents them interactively through pointing and clicking, and plays them back through various modes. No special animation or scripting skills are required.

EP 0 469 850 A2

BACKGROUND OF THE INVENTION

The present invention relates to pre-recording, editing and playing back presentations on a computer system. More particularly, this invention relates to editing screen session files, adding sound to and annotating such files, and playing back such files in a variety of ways. For purposes of describing the present invention, screen session files, recorded on disc storage devices in a personal computer system, may be metaphorically called a "tape" or "tapes". Similarly, as an extension of the metaphor, any part of such tapes may be referred to as a "clip" or "clips".

In the past, training or other information has been provided typically by in-person presentations or pre-recorded film or video tape. If provided by a live presenter, the audience loses potential advantages of "instant replay" and must attend the presentation at the convenience of the presenter or other members of the audience. If provided by film or video tape, the time and expense to produce high-quality film or tape presentations must be considered and budgeted, and special, often expensive presentation equipment - namely, film projector and screen or video tape player and television monitor - are required.

With the growing, wide-spread use of personal computer systems (hereafter also microcomputers or microcomputer systems) in industry, government, educational institutions and private homes, presentation of training or other information on such microcomputers is highly desireable. Often, such microcomputers are interconnected by a local area network (LAN) or wide area network (WAN), making the desirability of microcomputer-based, pre-recorded presentations even greater.

In conventional video, animation is achieved by displaying a sequence of changing pictures or frames several times a second. This creates the illusion of motion. In some systems for storing video data for use in computer systems, instead of storing entire frames, data representing changes from the previous frame are recorded. Thus, only the data required to change the previous frame into the next frame is saved. Much of the information that would remain the same between the frames is not stored which results in data compression. Here then, compression is simply removal of redundant information.

To edit conventional video sequences of frames are moved, added and/or deleted to produce an edited tape. With compressed video data, however, there are no frames but only data representing changes between frames, and editing cannot be done by merely rearranging the changes from frame-to-frame, since they are dependant on the previous image. Thus, the changes between frame, i.e. drawing commands, must be context sensitive, since the same command in a different context may not have the same result.

The most basic form of human communication is verbal. Therefore, sound is at least as important as text and graphics in a training tape. Sound not only conveys meaning, but also communicates compelling emotional content that might otherwise be absent. Since sound is not typically associated with computer presentations, addition of sound to such presentations is desireable for a totally effective presentation.

SUMMARY OF THE INVENTION

A system for pre-recording, editing, and playing back a presentation on a microcomputer constructed according to the present invention provides everything a user needs to create tapes, mark and edit them, record and synchronize sound on them, annotate them with buttons or other overlay graphics, present them interactively through pointing and clicking, and play them back through various modes. No special animation or scripting skills are required.

The types of presentations which may be prepared by using the present invention may be as widely varied as experienced users of microcomputers can make them. The most common types of training requirements are those where novices ask expert users questions such as "how to; "'what is;'and "show me". Using the present invention, experts may answer those questions with fully-produced tapes which can be viewed by any number of novice users, each at their own convenience. The present invention eliminates the need for expert users to repeat themselves over and over again. Of course, the tapes prepared or being prepared for individual use also can be presented on large screen computer monitors to groups or for management preview or final editing before release.

A presentation system for pre-recording, editing and play back of presentations on a computer system constructed according to the principles of the present invention comprises a recording component, a play back component and an editing component. The recording component records the desk top motions of the screen session of virtually any application program in real-time onto a tape stored in disc memory, and is available at any time as a desk accessory. The play back component provides several ways to play back tapes created in accordance with the present invention: 1) from within the editing component through the recording component; 2) from Hypercard using a special playback command; 3) via a special application designed for playing back tapes created using the present invention; or 4) by using the editing component to create tapes that become stand

Playback engine 102 includes elements from recording component 101 and sound component 104 which of no part of the present invention.

Editing component 103 edits the video portion of each tape according to the present invention by organizing video data into four parts as shown in Figure 2 and Tables I - V. The first part is clip list 201 which contains the numbered sequence of clips in the tape. Clip object 202 record contains information about an individual clip, such as its starting time and duration. Clip state record 203 which contains the drawing context at the start of a clip. Finally, clip drawing messages 204 include commands which are described in the Timbuktu Guides mentioned above together with additional ScreenRecorder specific messages that contain timing information and user events.

TABLE I

Clip List Record - List of clips in the tape.

Clip ID	First clip in tape
Clip ID	Second clip in tape
Clip ID	Third clip in tape
etc...	for each clip in tape

TABLE II

Clip Object Record - Information about an individual clip.

ID	the identifier to reference this clip
Length	Duration of clip in 60ths of a second
Starting Time	Relative starting time of clip in tape
Hard	True if clip has been moved from original location
Title	Name of clip enter by user
Small Picture	A small bit map of the Clip State bit map
State	Reference to Clip State
Data	Reference to Clip Data

TABLE III

Clip State Record - Context for the drawing commands.

BitMap	Initial Bit Map at the start of this clip
Width	Recorded screen width
Height	Recorded screen height
Graphports	Steven Graphports caches to save QDP Graphports

Message	Type	Data
Mouse Location	-1	Point
Timing Message	-4	Long Integer in ticks
Timing Base	-5	Long Integer in ticks
Mouse Down	-6	Point
Mouse Up	-7	Point
Key	-8	Char
Keyboard Modifiers	-9	Integer

Editing a tape

When a tape is first created it contains only one clip. In order to edit the tape, the tape must be divided into multiple clips. This is done by a process called marking. The user plays the tape starting from a clip, whenever the user comes to a point of interest, they issue a mark command that will divide the clip into two.

To divide a clip, it is necessary to create a new clip that contains the remainder of the drawing commands that have not played for the clip we are dividing. In order for the drawing commands to remain in the same context, it is necessary to get the current context from playback engine 102. The routine QDPGetState extracts the state information from the playback engine and places is in clip state record 203. Clip object record 202 being divided is then modified to reflect its shortened duration. Thus, two clips have been created which, when played back in sequence, will be visually the same as playing the original clip but they can now be moved around independent of each other. Clip dividing process 30 is desribed elsewhere in this specification in more detail with reference to Figure 3.

The clips can now be added, delete, or rearranged to edit the video tare. Clips reference in clip list record 201 can be rearranged or deleted using standard list manipulation methods. Clips may also be copied to other tapes by copying clip object records 202 and clip state records 203 and then their respective references added to clip list record 201.

Playing A tape

The present invention plays a tape one clip at a time in the sequence defined in clip list record 201. For each clip in the record, the proper graphic context is maintained and the appropriate drawing commands issued while regulating the playback speed with timing messages. This is done by first getting clip state record 201 and applying it to playback engine 102. This sets up the proper context for the drawing commands of this clip. The drawing commands are then returned to playback engine 102 at approximately the same rate as they where recorded. The timing information that was recorded along with clip drawing messages 204 regulates the playback speed. A time-base message, that provides the reference base for the other timing messages, is part of clip drawing messages 204. Timing messages are relative to this time base message.

Clip drawing messages 204 are played until the clip reaches the playback time in the message and then delayed until it is time to continue with more drawing commands. This is done until there are no more messages in the clip data stream. This process is described elsewhere in this specification in more detail with reference to Figures 4 and 5.

Maintaining Context

As mentioned elsewhere in this specification, clips may be played sequentially. In addition, any of the clips in the tape can be branched, since each clip has the entire drawing context to integrate into sequence of clips. Branching of clips allows for interactive presentations. Instead of playing a linear sequence through the tape, multiple paths through a tape may be controlled by the user.

All raster imaging systems have a state or context associated with them. At the least they have the bit-map for the image currently being displayed. In addition to the bit-map, most drawing packages have additional context associated with the drawing environment. In the case of QuickDraw, there are graph ports which provide such information as the current pen size and pattern.

includes MacRecorder, a hand-held sound input device, and SoundEdit, a program for recording and editing sampled and compressed sound data.

MacRecorder is more fully described in co-pending application for U.S. patent serial no. 07/291,808, entitled "SELF-POWERED DIGITAL TO ANALOG CONVERTER," filed December 29, 1988 which is assigned to the assignee of the present invention and incorporated by reference as if fully set forth herein. SoundEdit is more fully described in co-pending application for U.S. patent serial no. 07/302,510, entitled "COMPUTER SOUND EDITING DISPLAY SYSTEM", filed January 26, 1989 which is assigned to the assignee of the present invention and incorporated by reference as if fully set forth herein.

Reference is also made to the above mentioned Farallon Timbuktu User's Guide and the Farallon Timbuktu Remote User's Guide in the context of sound component 104.

Claims

1. In a system for pre-recording and playing back presentations on a computer system, apparatus for editing screen session data comprising:

storage means for storing user instructions and said screen session data, said screen session data comprising a plurality of frames of video bit-map data and related associated context commands, said video bit-map data representing at least one frame of screen session data and data representing successive changes from previous frames of said video bit-map data;

processing means, coupled to said storage means for processing said screen session data in response to said user instructions;

playback means, coupled to said storage means and said processing means for producing presentations in response to said user instructions, said presentations comprising successive frames of screen session data in response to said associated context commands; and

clip means, coupled to said storage means, processing means, and playback means, for successively dividing said frames of screen session data into clips comprising video bit-map data and associated context commands, and for editing said bit-map data and associated context commands in response to said user instructions.

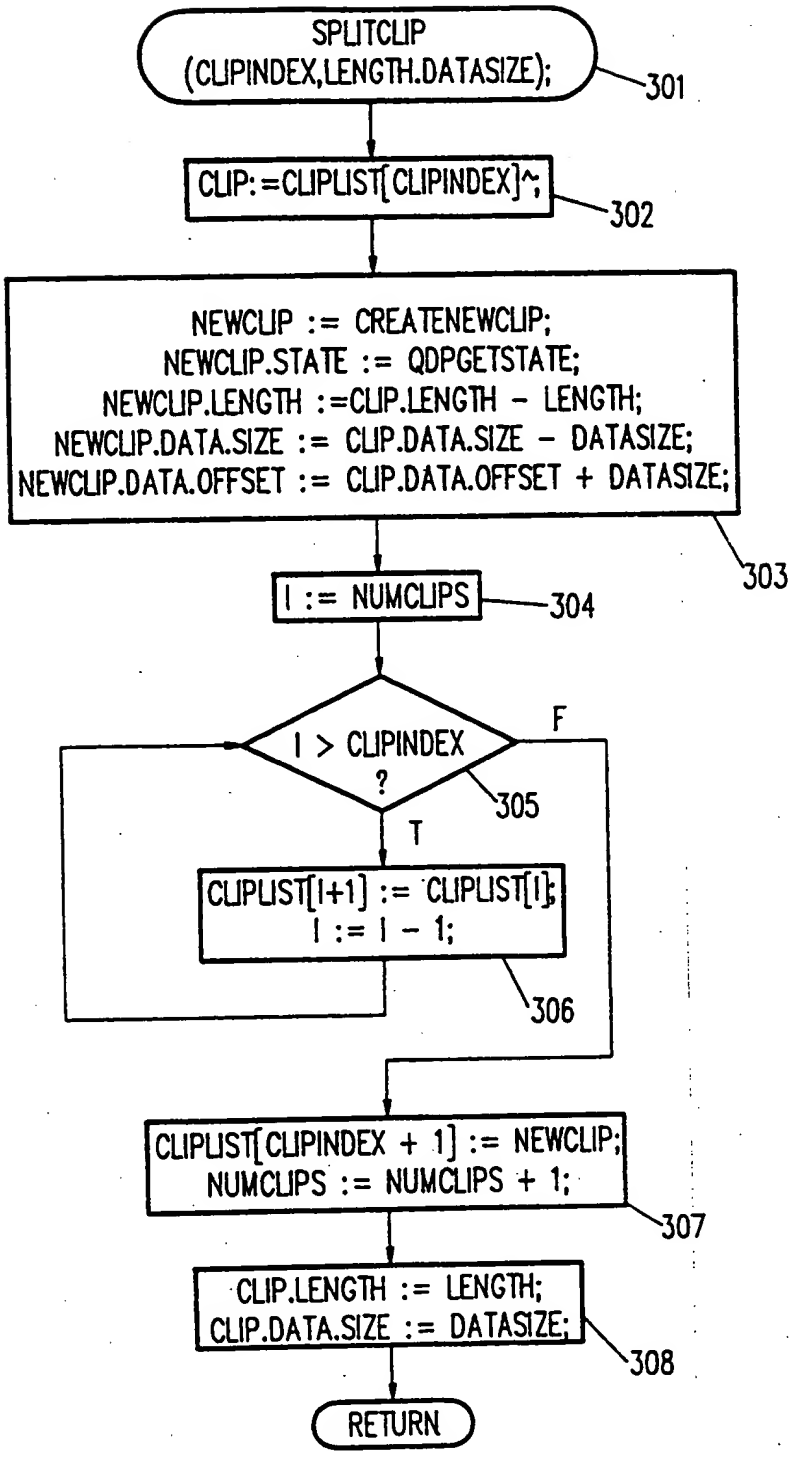
SPLITCLIP
(CLIPINDEX,LENGTH.DATASIZE);
301
CLIP:=CLIPLIST[CLIPINDEX]^;
302
NEWCLIP := CREATENEWCLIP;
NEWCLIP.STATE := QDPGETSTATE;
NEWCLIP.LENGTH :=CLIP.LENGTH - LENGTH;
NEWCLIP.DATA.SIZE := CLIP.DATA.SIZE - DATASIZE;
NEWCLIP.DATA.OFFSET := CLIP.DATA.OFFSET + DATASIZE;
303
I := NUMCLIPS
304
I > CLIPINDEX
?
305
F
T
CLIPLIST[I+1] := CLIPLIST[I];
I := I - 1;
306
CLIPLIST[CLIPINDEX + 1] := NEWCLIP;
NUMCLIPS := NUMCLIPS + 1;
307
CLIP.LENGTH := LENGTH;
CLIP.DATA.SIZE := DATASIZE;
308
RETURN

FIG. 3

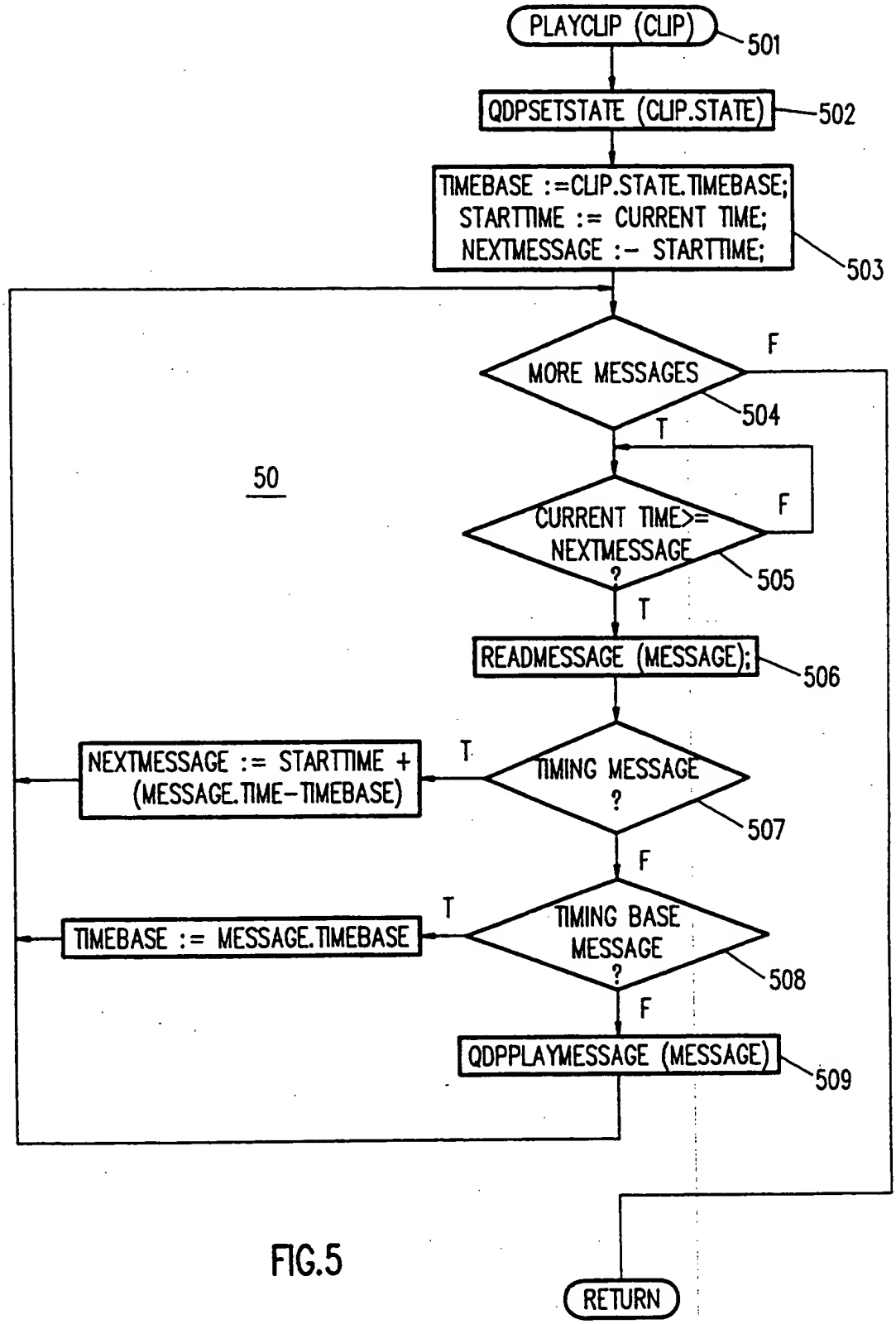
PLAYCLIP (CLIP)
501
QDPSETSTATE (CLIP.STATE)
502
TIMEBASE :=CLIP.STATE.TIMEBASE;
STARTTIME := CURRENT TIME;
NEXTMESSAGE :- STARTTIME;
503
MORE MESSAGES
504
F
T
50
CURRENT TIME>=
NEXTMESSAGE
?
505
F
T
READMESSAGE (MESSAGE);
506
NEXTMESSAGE := STARTTIME +
(MESSAGE.TIME-TIMEBASE)
T
TIMING MESSAGE
?
507
F
TIMEBASE := MESSAGE.TIMEBASE
T
TIMING BASE
MESSAGE
?
508
F
QDPPLAYMESSAGE (MESSAGE)
509
RETURN

FIG.5